IN THE CLAIMS:

Claims 1, 5, 6, 8, 11, 12, 17, 19, 22, 23, 28, 30, 33, 34, 38, 40-42, 45, 46, 51-53, and 56 are amended herein. Claims 7, 18, and 29 are cancelled. All pending claims and their present status are produced below.

1. (Currently Amended) A method comprising:

executing two separate instructions in a programmable processor, comprising:

executing a first instruction in [[a]]the programmable processor to set a rounding mode; and

executing a second instruction within the programmable processor to generate an integer result rounded according to the rounding mode, wherein the second instruction does not designate the rounding mode.

2. (Original) The method of claim 1, wherein executing the second instruction comprises executing an instruction that performs a rounded averaging operation.

3. (Original) The method of claim 1, wherein executing the second instruction comprises executing an instruction that performs a non-saturating, fixed-point fractional multiplication operation with rounding.

4. (Original) The method of claim 1, wherein executing the second instruction comprises executing an instruction that performs a right-shift operation with rounding.

5. (Currently Amended) The method of claim 1, wherein executing the first instruction comprises executing an instruction that sets a rounding mode selected from a group of rounding modes comprising:

rounding toward negative infinity;

rounding toward infinity;

rounding toward zero;

rounding away from zero;

rounding to a nearest integer, with a value of one-half being rounded toward~~toward~~ negative infinity;

rounding to a nearest integer, with a value of one-half being rounded toward~~toward~~ infinity;

rounding to a nearest integer, with a value of one-half being rounded toward~~toward~~ zero; and

rounding to a nearest integer, with a value of one-half being rounded away from zero.

6. (Currently Amended) A method comprising:

executing a first instruction to set a rounding mode;

performing an operation within a programmable processor to produce a result according to a second instruction, wherein the second instruction does not designate the rounding mode;

adding a rounding term to the result to obtain an intermediate result, the rounding term determined at least in part as a function of [[a]]the rounding mode[[,]] and a shift amount~~, and a sign of the result of the operation~~; and

right-shifting the intermediate result by the shift amount.

7. (Canceled)

8. (Currently Amended) The method of claim [[7]]6, wherein executing the first instruction comprises executing an instruction that sets a rounding mode selected from the group of rounding modes comprising:

rounding toward~~toward~~ negative infinity;

rounding toward infinity;

rounding toward zero;

rounding away from zero;

rounding to a nearest integer, with a value of one-half being rounded toward negative infinity;

rounding to a nearest integer, with a value of one-half being rounded toward infinity;

rounding to a nearest integer, with a value of one-half being rounded toward zero; and

rounding to a nearest integer, with a value of one-half being rounded away from zero.

9. (Original) The method of claim 6, wherein the operation is a rounded averaging operation of two or four unsigned byte vectors.

10. (Original) The method of claim 6, wherein the operation is a non-saturating fixed-point fractional multiplication operation with rounding of a set of vector operands selected from signed half-word vectors, unsigned half-word vectors, signed word vectors and unsigned word vectors.

11. (Currently Amended) The method of claim [[6]]1, wherein the ~~operation is~~second instruction includes an arithmetic right-shift operation on a vector operand by an immediate shift amount with rounding, and wherein the vector operand comprises a signed byte vector, an unsigned byte vector, a signed double word or an unsigned double word.

12. (Currently Amended) A method of compiling a processor-readable software program comprising parsing [[the]]a software program to produce instructions executable by a programmable processor, wherein the instructions include two separate instructions, comprising a first instruction that sets a rounding mode, and a second instruction that performs an arithmetic operation yielding an integer result rounded according to the rounding mode, wherein the second instruction does not designate the rounding mode.

13. (Original) The method of claim 12, wherein the second instruction performs a rounded averaging operation.

14. (Original) The method of claim 12, wherein the second instruction performs a non-saturating fixed-point fractional multiplication operation with rounding.

15. (Original) The method of claim 12, wherein the second instruction performs a right-shift operation with rounding.

16. (Original) The method of claim 12, wherein the first instruction sets a rounding mode selected from the group of rounding modes comprising:

rounding toward negative infinity;

rounding toward infinity;

rounding toward zero;

rounding away from zero;

rounding to a nearest integer, with a value of one-half being rounded toward negative infinity;

rounding to a nearest integer, with a value of one-half being rounded toward infinity;

rounding to a nearest integer, with a value of one-half being rounded toward zero; and

rounding to a nearest integer, with a value of one-half being rounded away from zero.

17. (Currently Amended) A method of compiling a processor-readable software program comprising parsing [[the]]a software program to produce instructions executable by a programmable processor, wherein the instructions cause the programmable processor to:

execute a first instruction to set a rounding mode;

perform an arithmetic operation according to a second instruction, wherein the second instruction does not designate the rounding mode;

add a rounding term to a result of the arithmetic operation to obtain an intermediate result, the rounding term determined at least in part as a function of [[a]]the rounding mode[[,]]and a shift amount~~, and a sign of the result of the arithmetic operation~~; and

right-shift the intermediate result by the shift amount.

18. (Canceled)

19. (Currently Amended) The method of claim [[18]]17, wherein executing the first instruction comprises executing an instruction that sets a rounding mode selected from the group of rounding modes comprising:

rounding toward negative infinity;

rounding toward infinity;

rounding toward zero;

rounding away from zero;

rounding to a nearest integer, with a value of one-half being rounded toward negative infinity;

rounding to a nearest integer, with a value of one-half being rounded toward infinity;

rounding to a nearest integer, with a value of one-half being rounded toward zero; and

rounding to a nearest integer, with a value of one-half being rounded away from zero.

20. (Original) The method of claim 17, wherein the arithmetic operation is a rounded averaging operation of two or four unsigned byte vectors.

21. (Original) The method of claim 17, wherein the arithmetic operation is a non-saturating fixed-point fractional multiplication operation with rounding of a set of vector operands selected from signed half-word vectors, unsigned half-word vectors, signed word vectors and unsigned word vectors.

22. (Currently Amended) The method of claim [[17]]12, wherein the arithmetic operation is an arithmetic right-shift operation on a vector operand by an immediate shift amount with rounding, wherein the vector operand comprises a signed byte vector, an unsigned byte vector, a signed double word or an unsigned double word.

23. (Currently Amended) A processor-readable medium having processor-executable instructions for:

executing two separate instructions in a programmable processor, comprising:

executing a first instruction in [[a]]the programmable processor to set a rounding mode; and

executing a second instruction within the programmable processor to generate an integer result rounded according to the rounding mode, wherein the second instruction does not designate the rounding mode.

24. (Original) The processor-readable medium of claim 23, wherein executing the second instruction comprises executing an instruction that performs a rounded averaging operation.

25. (Original) The processor-readable medium of claim 23, wherein executing the second instruction comprises executing an instruction that performs a non-saturating fixed-point fractional multiplication operation with rounding.

26. (Original) The processor-readable medium of claim 23, wherein executing the second instruction comprises executing an instruction that performs a right-shift operation with rounding.

27. (Original) The processor-readable medium of claim 23, wherein executing the first instruction comprises executing an instruction that sets a rounding mode selected from the group of rounding modes comprising:

rounding toward negative infinity;

rounding toward infinity;

rounding toward zero;

rounding away from zero;

rounding to a nearest integer, with a value of one-half being rounded toward negative infinity;

rounding to a nearest integer, with a value of one-half being rounded toward infinity;

rounding to a nearest integer, with a value of one-half being rounded toward zero; and

rounding to a nearest integer, with a value of one-half being rounded away from zero.

28. (Currently Amended) A processor-readable medium having processor-executable instructions for:

executing a first instruction to set a rounding mode;

performing an arithmetic operation according to a second instruction, wherein the second instruction does not designate the rounding mode;

adding a rounding term to a result of the arithmetic operation to obtain an intermediate result, the rounding term determined at least in part as a function of [[a]]the rounding mode[[,]]and a shift amount~~, and a sign of the result of the arithmetic operation~~; and

right-shifting the intermediate result by the shift amount.

29. (Canceled)

30. (Currently Amended) The processor-readable medium of claim [[29]]28, wherein executing the first instruction comprises executing an instruction that sets a rounding mode selected from the group of rounding modes comprising:

rounding toward negative infinity;

rounding toward infinity;

rounding toward zero;

rounding away from zero;

rounding to a nearest integer, with a value of one-half being rounded toward negative infinity;

rounding to a nearest integer, with a value of one-half being rounded toward infinity;

rounding to a nearest integer, with a value of one-half being rounded toward zero; and

rounding to a nearest integer, with a value of one-half being rounded away from zero.

31. (Original) The processor-readable medium of claim 28, wherein the arithmetic operation is a rounded averaging operation of two or four unsigned byte vectors.

32. (Original) The processor-readable medium of claim 28, wherein the arithmetic operation is a non-saturating fixed-point fractional multiplication operation with rounding of a set of vector operands selected from signed half-word vectors, unsigned half-word vectors, signed word vectors and unsigned word vectors.

33. (Currently Amended) The processor-readable medium of claim [[28]]23, wherein the ~~arithmetic operation is~~second instruction includes an arithmetic right-shift operation on a vector operand by an immediate shift amount with rounding, wherein the vector operand comprises a signed byte vector, an unsigned byte vector, a signed double word or an unsigned double word.

34. (Currently Amended) A processor, comprising:

a control register to store a rounding mode of a first instruction;

a functional unit; and

a control unit to direct the functional unit to perform an arithmetic function according to the rounding mode in response to a second instruction, wherein the second instruction does not designate the rounding mode.

35. (Original) The processor of claim 34, wherein the second instruction comprises a rounded averaging operation.

36. (Original) The processor of claim 34, wherein the second instruction comprises a non-saturating fixed-point fractional multiplication operation with rounding.

37. (Original) The processor of claim 34, wherein the second instruction comprises a right-shift operation with rounding.

38. (Currently Amended) The processor of claim 34, further comprising:

a fetch unit configured to receive an instruction from an instruction stream;

a decode unit configured to decode the received instruction; and

a register file coupled to the plurality of functional units and configured to store an~~the integer~~ result.

39. (Original) The processor of claim 34, wherein the first instruction sets a rounding mode selected from the group of rounding modes comprising:

rounding toward negative infinity;

rounding toward infinity;

rounding toward zero;

rounding away from zero;

rounding to a nearest integer, with a value of one-half being rounded toward negative infinity;

rounding to a nearest integer, with a value of one-half being rounded toward infinity;

rounding to a nearest integer, with a value of one-half being rounded toward zero; and

rounding to a nearest integer, with a value of one-half being rounded away from zero.

40. (Currently Amended) A processor, comprising:

a control unit comprising a control register configured to store a representation of a selected rounding mode designated by a first instruction;

at least one functional unit coupled to the control register;

a fetch unit configured to receive a[[n]]second instruction from an instruction stream;

a decode unit configured to decode the ~~received~~second instruction; and

a register file coupled to the plurality of functional units,

the control unit configured to

perform an arithmetic operation according to the second instruction, wherein the second instruction does not designate the rounding mode,

add a rounding term to a result of the arithmetic operation to obtain an intermediate result, the rounding term determined at least in part as a function of the selected rounding mode[[,]]and a shift amount~~, and a sign of the result of the arithmetic operation~~,

right-shift the intermediate result by the shift amount to generate a rounded result, and

store the rounded result in the register file.

41. (Currently Amended) The processor of claim 40, wherein the control unit is further configured to execute [[an]]the first instruction [[that]]to set[[s]] the rounding mode.

42. (Currently Amended) The processor of claim 41, wherein the first instruction sets a rounding mode selected from the group of rounding modes comprising:

rounding toward negative infinity;

rounding toward infinity;

rounding toward zero;

rounding away from zero;

rounding to a nearest integer, with a value of one-half being rounded toward negative infinity;

rounding to a nearest integer, with a value of one-half being rounded toward infinity;

rounding to a nearest integer, with a value of one-half being rounded toward zero; and

rounding to a nearest integer, with a value of one-half being rounded away from zero.

43. (Original) The processor of claim 40, wherein the arithmetic operation is a rounded averaging operation of two or four unsigned byte vectors.

44. (Original) The processor of claim 40, wherein the arithmetic operation comprises performing a non-saturating fixed-point fractional multiplication operation with rounding of a set of vector operands selected from signed half-word vectors, unsigned half-word vectors, signed word vectors and unsigned word vectors.

45. (Currently Amended) The processor of claim [[40]]34, wherein the arithmetic operation is an arithmetic right-shift operation on a vector operand by an immediate shift

amount with rounding, wherein the vector operand comprises a signed byte vector, an unsigned byte vector, a signed double word or an unsigned double word.

46. (Currently Amended) A system comprising:
a memory; and
a processor comprising
a control register to store a rounding mode of a first instruction,
a functional unit, and
a control unit to direct the functional unit to perform an arithmetic function according to the rounding mode in response to a second instruction, wherein the second instruction does not designate the rounding mode.

47. (Original) The system of claim 46, wherein the second instruction comprises a rounded averaging operation.

48. (Original) The system of claim 46, wherein the second instruction comprises a non-saturating fixed-point fractional multiplication operation with rounding of a set of vector operands selected from signed half-word vectors, unsigned half-word vectors, signed word vectors and unsigned word vectors.

49. (Original) The system of claim 46, wherein the second instruction comprises a right-shift operation with rounding.

50. (Original) The system of claim 46, wherein the rounding mode is selected from the group of rounding modes comprising:
rounding toward negative infinity;
rounding toward infinity;
rounding toward zero;
rounding away from zero;
rounding to a nearest integer, with a value of one-half being rounded toward negative infinity;
rounding to a nearest integer, with a value of one-half being rounded toward infinity;

rounding to a nearest integer, with a value of one-half being rounded toward zero; and

rounding to a nearest integer, with a value of one-half being rounded away from zero.

51. (Currently Amended) A system, comprising:

a memory; and

a control unit comprising a control register configured to store a representation of a selected rounding mode designated by a first instruction;

at least one functional unit coupled to the control register;

a fetch unit configured to receive a[[n]]second instruction from an instruction stream;

a decode unit configured to decode the ~~received~~second instruction; and

a register file coupled to the plurality of functional units,

the control unit configured to

perform an arithmetic operation according to the second instruction, wherein the second instruction does not designate the rounding mode,

add a rounding term to a result of the arithmetic operation to obtain an intermediate result, the rounding term determined at least in part as a function of the selected rounding mode[[,]]and a shift amount~~, and a sign of the result of the arithmetic operation~~,

right-shift the intermediate result by the shift amount to generate a rounded result, and

store the rounded result in the register file.

52. (Currently Amended) The system of claim 51, wherein the processor is further configured to execute [[an]]the first instruction [[that]]to set[[s]] the rounding mode.

53. (Currently Amended) The system of claim 52, wherein executing the first instruction comprises executing an instruction that sets a rounding mode selected from the group of rounding modes comprising:

rounding toward negative infinity;

rounding toward infinity;

rounding toward zero;

rounding away from zero;

rounding to a nearest integer, with a value of one-half being rounded toward negative infinity;

rounding to a nearest integer, with a value of one-half being rounded toward infinity;

rounding to a nearest integer, with a value of one-half being rounded toward zero; and

rounding to a nearest integer, with a value of one-half being rounded away from zero.

54. (Original) The system of claim 51, wherein the arithmetic operation is a rounded averaging operation of two or four unsigned byte vectors.

55. (Original) The system of claim 51, wherein the arithmetic operation is a non-saturating fixed-point fractional multiplication operation with rounding of a set of vector operands selected from signed half-word vectors, unsigned half-word vectors, signed word vectors and unsigned word vectors.

56. (Currently Amended) The system of claim [[51]]46, wherein the ~~arithmetic operation is~~second instruction includes an arithmetic right-shift operation on a vector operand by an immediate shift amount with rounding, wherein the vector operand comprises a signed byte vector, an unsigned byte vector, a signed double word or an unsigned double word.